A Forecasting and Inventory Model for Short Lifecycle Products with Seasonal Demand Patterns

by

Wesley D. Margeson
Bachelor of Science in Mechanical Engineering and
Materials Science, Duke University (1995)

Submitted to the Department of Mechanical Engineering and the Sloan School of Management in Partial Fulfillment of the Requirements for the Degrees of

Master of Science in Management
and
Master of Science in Mechanical Engineering

In Conjunction with the Leaders for Manufacturing Program
at the Massachusetts Institute of Technology
June 2003

Signature of Author____________
Department of Mechanical Engineering
Sloan School of Management
May 2003

Certified by____________
Donald B. Rosenfield, Thesis Supervisor
Director, Leaders for Manufacturing Fellows Program
Senior Lecturer, Sloan School of Management

Certified by____________
James M. Masters, Thesis Supervisor
Executive Director, Masters of Engineering in Logistics Program
Engineering Systems Division

Certified by____________
Stanley B. Gershwin, Thesis Reader
Associate Director, Lab for Manufacturing and Productivity
Senior Research Scientist, Department of Mechanical Engineering

Accepted by____________
Margaret Andrews, Executive Director of Masters Program
Sloan School of Management

Accepted by____________
Ain Sonin, Chairman, Graduate Committee
Department of Mechanical Engineering

This page is intentionally left blank.

A Forecasting and Inventory Model for Short Lifecycle Products with Seasonal Demand Patterns

by

Wesley D. Margeson

Submitted to the Sloan School of Management and the Department of Mechanical Engineering on May 9, 2003 in partial fulfillment of the Requirements for the Degrees of

Master of Science in Management and
Master of Science in Mechanical Engineering

Abstract

The lifecycles of many products are becoming shorter and shorter as innovation and time-to-market become key aspects of modern corporations' success in the marketplace. Few methods for inventory management exist that are capable of contending with product lifecycles measured in months rather than years. This situation is especially acute in the semiconductor industry, where a single generation of a product typically spends less than 6 months in the marketplace - demand for these products is therefore highly non-stationary and stochastic. Supply chain management is further complicated by non-stationary, stochastic lead times and the necessity for complex demand forecasting procedures - the use of time-series forecasting models is therefore generally infeasible for these products.

We present a new method in which exogenous short and medium-term forecasts are combined with a new method of exponential smoothing to generate replenishment forecasts for finished goods inventories. This method is designed to contend with highly non-stationary, stochastic demand and lead time patterns, daily seasonal effects, unusual probability distributions, capacity constraints, and short product lifecycles across a variety of products. Historical simulations of the model for many of Intel Corporation's Central Processing Unit products reveal dramatic inventory reductions compared to the ad hoc inventory management policies presently in use.

Thesis Supervisor: Donald Rosenfield
Senior Lecturer, Sloan School of Management
Director, Leaders for Manufacturing Fellows Program

Thesis Supervisor: James Masters
Executive Director, Masters in Logistics Program

This page is intentionally left blank.

Acknowledgements

No words can express my gratitude to my wife, Jennifer, who has been my staunchest supporter throughout these two years at MIT. Her love has been the only constant during this experience in which the "birth" of this thesis was paralleled by the birth of our first child, Lillian Paige. I hold both of them dearest to my heart.

I would like to thank the Leaders for Manufacturing (LFM) Program and Intel Corporation for providing me with the opportunity to grow not only professionally, but also socially and intellectually – I am a better person for my experiences through this unique program. Many thanks also to my classmates in the LFM Program, who have given me friendship and taught me much about how to lead, learn, and live to the fullest of my abilities. Special thanks to Lou Rassey, without whom I could not have gone nearly as far as I did with this project and who has helped me to grow as a leader through counsel, friendship, and example.

I am greatly indebted to Tony Newlin and Michael Waithe for their support and supervision of this project, which would not have succeeded without Tony to champion it or Michael's willingness to allow me to undertake a project of such import.

I would also like to thank Intel's Supply Network Group for a terrific research fellowship experience. Special thanks to Michael Donnelly and Eric Johnson for their encouragement and assistance over the course of the project. Thanks to Karl Kempf for his willingness to listen to and challenge my ideas. Thanks also to Michelle Silver and Tom Maher for their belief in this project and help in the final stages.

I am grateful for the help of my thesis advisors, Donald Rosenfield and James Masters. I am most indebted to Don, who gave so much of his time as well as patient counsel over the course of the project. His advice helped me to break down many barriers that I could not have otherwise overcome. Thanks also to Jonathan Byrnes, whose teachings gave me the inspiration and vision I needed to undertake this project.

Finally, I would like to thank my parents, Jill Margeson and Don Margeson for their love and support.

This page is intentionally left blank.

Table of Contents

Table of Figures

1 INTRODUCTION

The objective of this thesis is to establish a forecasting and inventory replenishment model capable of contending with highly non-stationary, stochastic demand and lead-seasonal effects, unusual probability distributions, and short product lifecycles across a variety of products. More specifically, this model addresses the need for generation of daily forecasts of demand from other forecasting techniques that are considered exogenous to the model. The thesis includes an analysis of supply chain requirements, a quantitative model for daily demand and inventory forecasts, and an assessment of organizational and strategic alignment to the new system. This research was conducted at Intel Corporation's Supply Network Group from June through December of 2003.

1.1 INDUSTRY AND COMPANY BACKGROUND

Throughout the course of its history, Intel has derived its success largely from the pursuit of a single strategy: innovation. This strategy is a necessary one in an industry that has its own law to describe the pace of technological advance: namely, Moore's Law[1]. Extremely high capital costs, cyclical markets, and cutting-edge manufacturing technology characterize the semiconductor industry. Intel Corporation manufactures and distributes a wide variety of semiconductor products, including processors, chipsets, flash memory, communications products, and motherboards.

[1] Gordon Moore, a co-founder of Intel Corporation, predicted more than 20 years ago that the number of transistors that could be placed on a given silicon chip would double every eighteen months. This maxim has proved largely true and in many ways has become the implicit goal of Intel, its suppliers, and its competitors in the semiconductor industry.

Intel's relentless quest to improve itself and its products has resulted in a dominant position in many of the market segments in which it competes, including an 80% worldwide market segment share in CPUs. Holding to the course of Moore's Law has brought Intel tremendous revenue growth over the past decade, in addition to higher market segment share in CPUs. However, the price of the technology required to produce ever-smaller transistors on its chips has increased as well. The current total cost to build a wafer fabrication facility has risen to more than $2 billion.

These high fixed costs understandably make up a large percentage of the cost of producing a chip. This is rather intuitive when one compares the price of the chemicals and silicon that the chip is made of to the cost of the factory and tools that made it. For a product such as a CPU with low variable costs and high market value, the opportunity cost of capital lost through excess inventories is perceived by many to be less than the cost of lost sales. Additionally, the rapid pace of product and manufacturing process improvement at Intel sometimes has resulted in periods where demand outstrips supply. The ensuing stockouts result in lost revenue for Intel and if prolonged, lost market share. The result is that high inventory levels at Intel have become accepted as a necessary evil to avoid loss of market share and revenue – an attitude that must be overcome for inventory reduction to become possible.

This attitude is may seem surprising when one considers the pace of price erosion and obsolescence experienced by many of Intel's products. However, it is important to note that the cost of holding inventory is determined by the variable cost to produce it, not its market price. Inventory write-downs are only driven by market price when that price is less than variable cost.

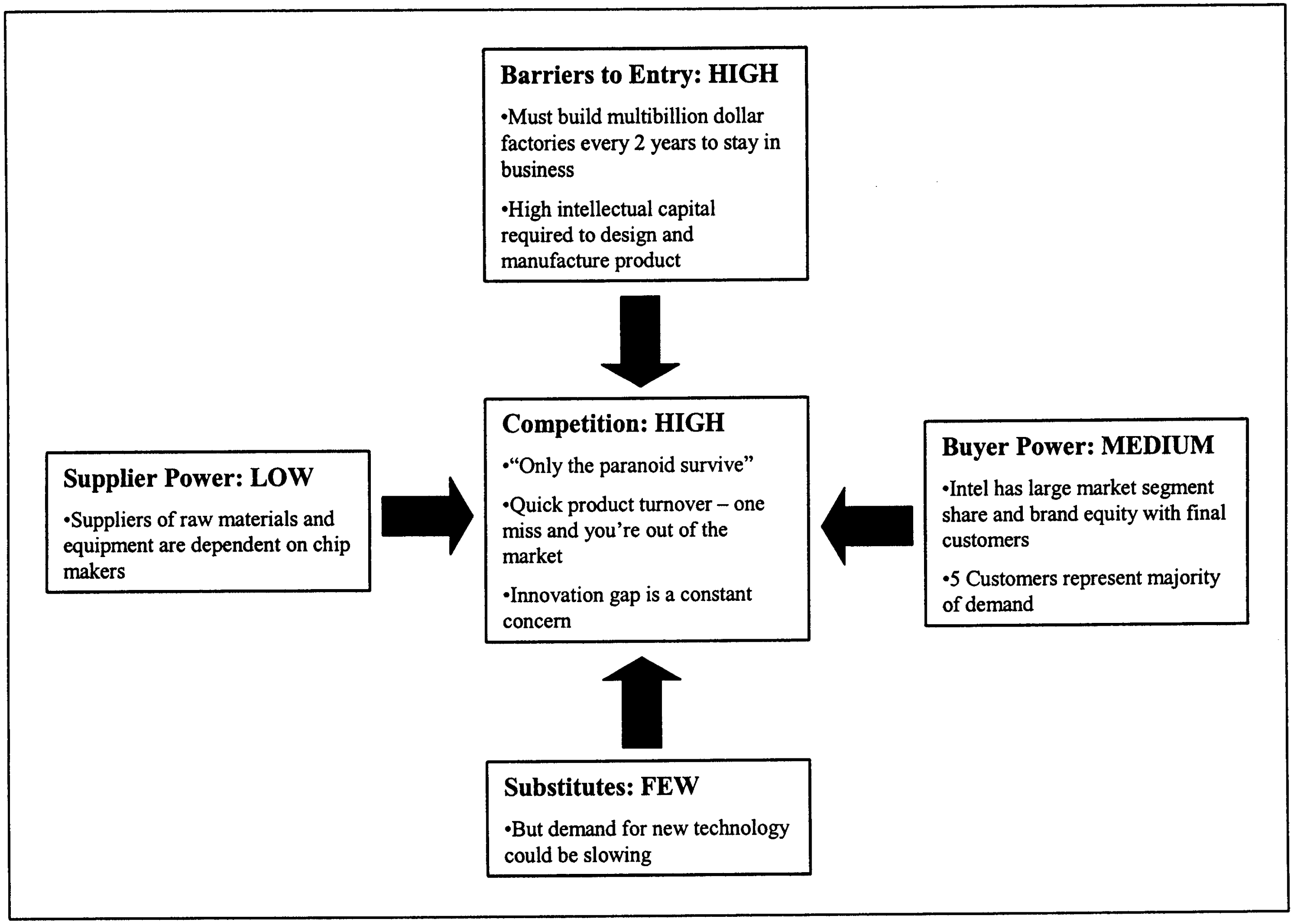
Recently the semiconductor industry has been experiencing a downturn longer than any in its entire history. Intel is seeking new ways to cut costs in order to maintain the high level of profitability it has sustained in the past. These economic pressures provide some of the motivation for change necessary to overcome past acceptance of high cost inventory management practices.

A high level strategic analysis of Intel is depicted in Figure 1-1[2]. This analysis, while static, illustrates how the changing landscape of a dynamic industry provides an additional source of motivation for better inventory control systems. This motivation stems from the combination of the uncertain future of product substitutes for microprocessors with the highly competitive nature of the industry.

A recent article in the Harvard Business Review [Christensen et al, 1999] suggests a far less profitable future for Intel's star product – high power CPUs. For most of the history of the semiconductor industry, the dominant factor for success in the CPU marketplace has been product performance. This article suggests that as the performance of existing products more closely matches the needs of customers, additional factors become more important. These factors include product availability, delivery, price, and features.

[2] Michael E. Porter first presented this type of analysis in 1979 in the Harvard Business Review [Porter, 1979].

Figure 1-1. Strategic Analysis of the Microprocessor Industry



Barriers to Entry: HIGH

- Must build multibillion dollar factories every 2 years to stay in business
- High intellectual capital required to design and manufacture product

Supplier Power: LOW

- Suppliers of raw materials and equipment are dependent on chip makers

Competition: HIGH

- “Only the paranoid survive”
- Quick product turnover – one miss and you’re out of the market
- Innovation gap is a constant concern

Buyer Power: MEDIUM

- Intel has large market segment share and brand equity with final customers
- 5 Customers represent majority of demand

Substitutes: FEW

- But demand for new technology could be slowing

The implementation of our model offers Intel the ability to meet these new requirements by lowering costs without sacrificing product availability. Lower costs would allow Intel to increase profits (or offer lower prices, should it become necessary to do so). In addition, an inventory replenishment model gives Intel the ability to understand the relationship between service level, inventories, and capacity allocation. This understanding will help Intel to make more informed decisions that will allow it to become more profitable.

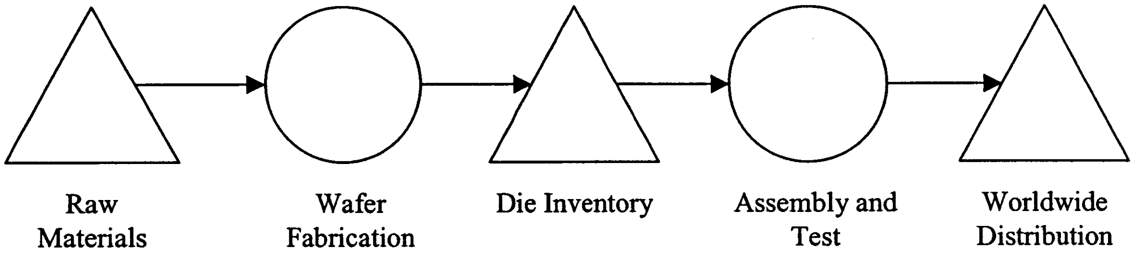
1.2 Supply Chain Overview and Challenges

The task of supply chain management faced by Intel is highly complex, characterized by a long internal supply network, thousands of products, short product lifecycles, and worldwide operations. In this section we examine the supply chain management challenge in more detail and place boundaries on the scope of the problem we will address in this thesis.

For the development of our model, we have chosen to focus on Central Processing Units (CPUs), the company's most well known and most profitable product. CPUs represent the majority of Intel's production volume and cost-of-goods-sold, and therefore hold the greatest opportunity for reduction in supply chain costs.

Intel's supply chain is characterized by two major manufacturing stages followed by a finished goods distribution system. Manufacturing and distribution are conducted on a global scale across dozens of factories and warehouses with hundreds of customers. Figure 1-2 depicts Intel's internal supply chain at a high level. The lead times of both production stages are stochastic and sufficiently long (weeks to months) that production is performed on a make-to-forecast basis, with the bulk of the total lead time residing in the first production stage.

Figure 1-2. Supply Chain for CPU Production



CPUs are distributed to two primary classes of customers: resellers and computer manufacturers, with the bulk of demand falling in the latter category. The reseller channel requires one additional packaging step prior to shipment – this step currently pulls its materials from finished goods inventories rather than ordering them from the assembly/test factories. Distribution is performed mainly through air freight directly from one of three factory distribution centers to designated customer pickup points such as cross docks, although Intel does own and operate some regional warehouses to serve its customers' last-minute requests. For our purposes, inventory at the three factory distribution centers can be considered together because customer demand for a given product is equally likely to be served from any distribution center.

The number of line items produced from a common set of raw materials is very large, and the production quantities of each line item are stochastic and evolutionary in nature. The hierarchy of product definition at the finished goods level is shown in Figure 1-3.

Figure 1-3. Product Definition Hierarchy

M Product Types (chipset, microprocessor, etc.)
X
N Product or Process Revisions (Step A, Step B, etc.)
X
W Product Families (type of core, cache size)
X
X Assembly Configurations (pin package)
X
Y Speeds (1.8 GHz, 2.0 GHz, etc.)
X
Z Packaging Options (individual box, bulk, etc.)
=
Thousands of possible line items!!!

In addition, the lifetime of a line item averages a mere six to nine months. This short lifecycle is the result of the generational nature of Intel's products. A product marketed to the end consumer as a 2.0 GHz Pentium IV may have a market life of approximately 18 months. However, during this time there are several reductions in the size of the chip, called "steppings." These steppings allow Intel to increase the number of chips produced on a silicon wafer and reduce their fabrication costs proportionally. One stepping replaces another, although there is overlap between them due to the needs of Intel's customers. These facts drastically complicate the management of finished goods inventories over the product lifecycle. To summarize the supply chain challenge before us:

- Intel's supply chain is not a simple chain but a complex network.
- Demand and production characteristics are non-stationary and stochastic in nature.

- The total number of line items managed at the finished goods level numbers in the thousands.

In order to limit the problem before us, we have narrowed our focus to the management of finished goods inventories for CPUs. However, with this initial model we have developed the building blocks of a system that can be expanded to include additional product lines and other parts of Intel's supply chain.

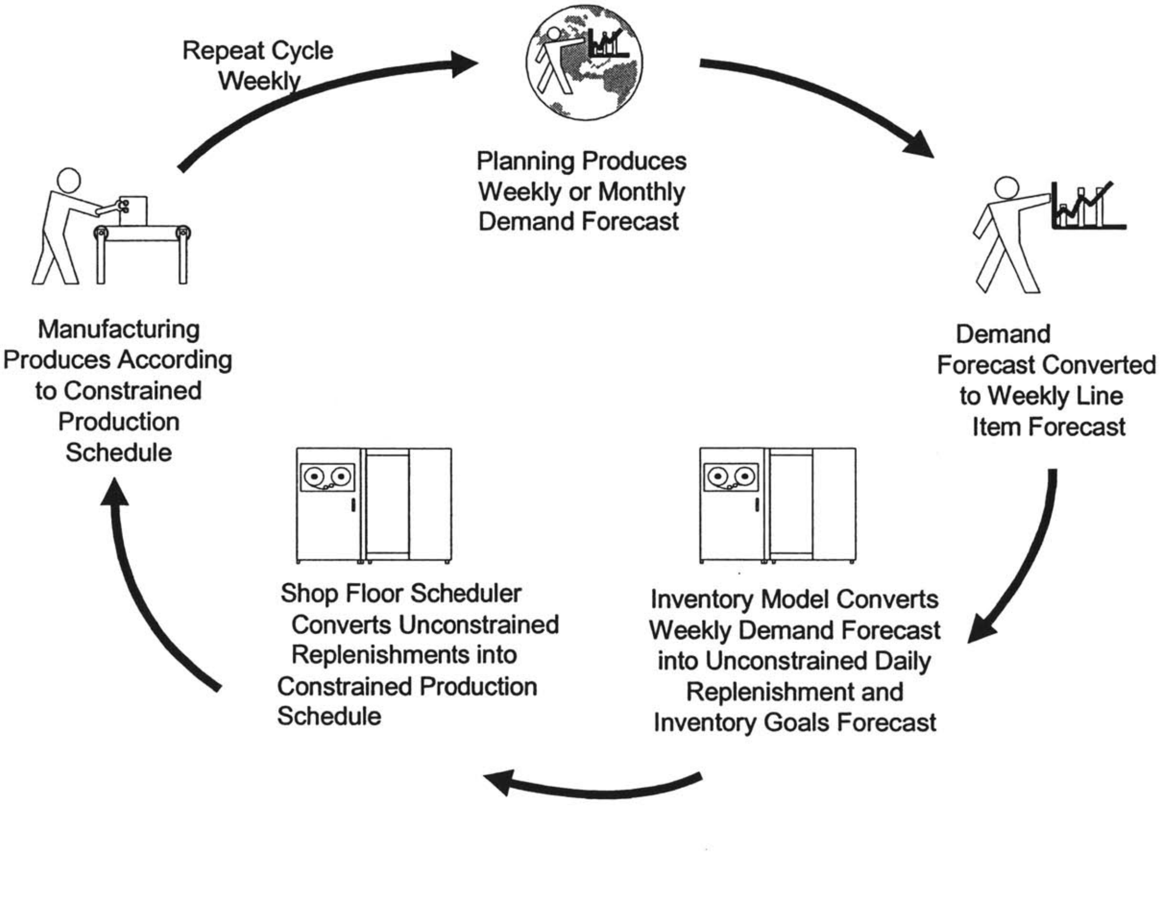
We have chosen finished goods inventories (FGI) because they make up the bulk of the inventory carrying costs for Intel. Additionally, FGI represents an important transfer of responsibility from planning and manufacturing to logistics. Bridging this organizational gap first would hold significant symbolic and literal meaning as the beginning of a new paradigm in supply chain management at Intel.

1.3 PROJECT APPROACH AND OUTCOMES

This thesis is part of a wider effort that seeks to begin a paradigm shift in Intel's supply chain management practices by converting from a manual push system (build to forecast) to an automated pull system (replenish to forecast). Our method will add statistical methods to experience in order to bring about dramatic reductions in supply chain costs through the reduction of finished goods inventories and inventory and production planning efforts. These methods are designed to contend with non-stationary, stochastic demand and lead time patterns, weekly seasonal effects (we use the term seasonal to denote day-of-the-week effects), unusual probability distributions, and short product lifecycles across a variety of products.

Our approach can be broken down into two major components - a forecasting model and an inventory replenishment model. The forecasting model is a variation on exponential smoothing. Demand data are first differenced by a 7-day moving average to capture day-of-the-week effects. These effects are then an exponentially smoothed to obtain day-of-the-week factors. Optimization is used to obtain the seasonal smoothing coefficient that minimizes the squared seasonal error. The result is a set of multiplicative factors for error and day-of-the-week that can be paired with Intel's weekly demand forecasts to estimate daily demand and error over the forecast horizon. In addition, estimates of lead times, lead time variability, and Intel's forecast error are obtained through exponential smoothing or other methods. Figure 1-4 illustrates how the model would fit with Intel's planning and production processes.

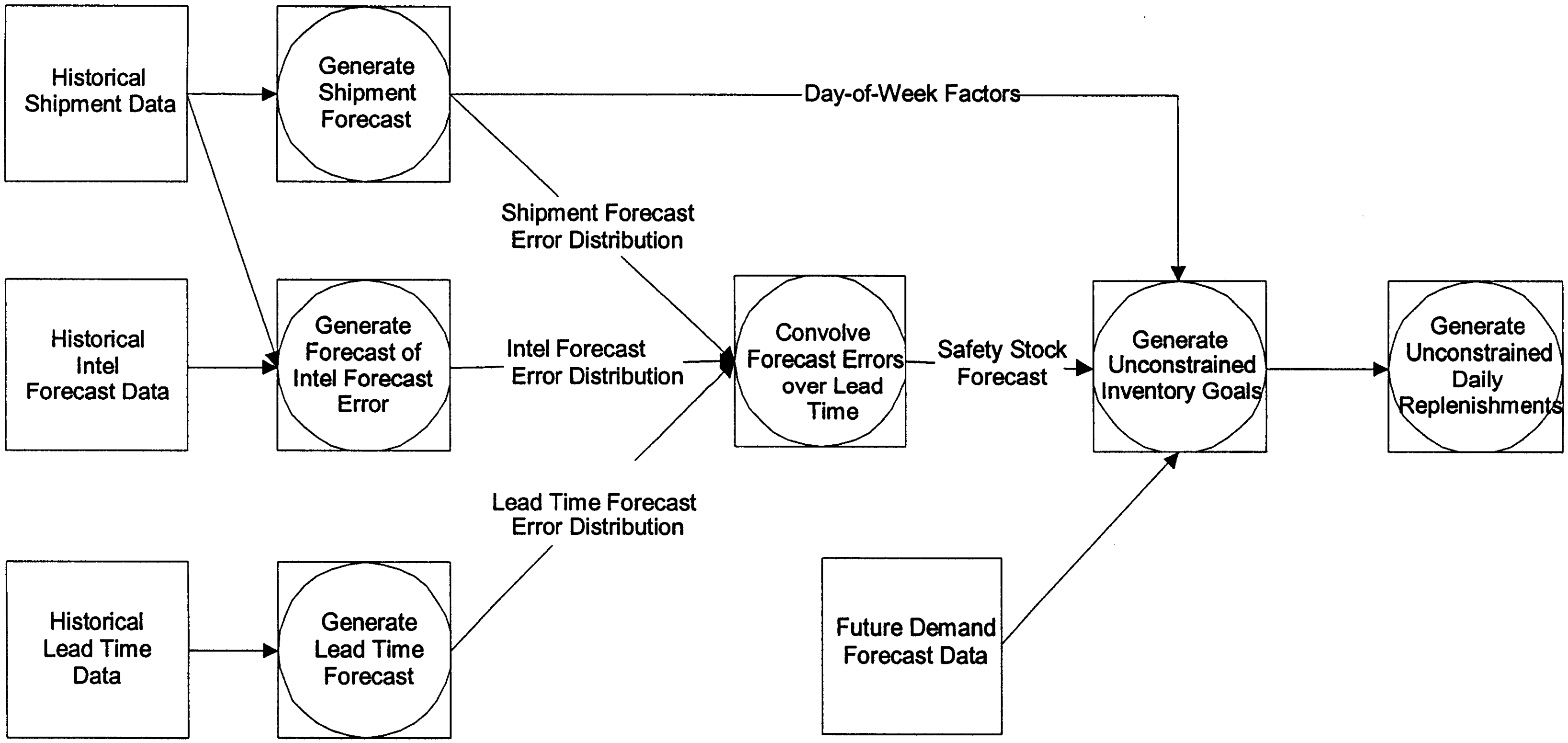
Figure 1-4. Weekly to Daily Production Plan Conversion


Repeat Cycle Weekly
Planning Produces Weekly or Monthly Demand Forecast
Demand Forecast Converted to Weekly Line Item Forecast
Inventory Model Converts Weekly Demand Forecast into Unconstrained Daily Replenishment and Inventory Goals Forecast
Shop Floor Scheduler Converts Unconstrained Replenishments into Constrained Production Schedule
Manufacturing Produces According to Constrained Production Schedule

For the inventory model, the vectors for demand, lead time, and forecast error are then convolved over the forecasted lead time to produce estimates of lead time demands over the forecast horizon. The convolved error vector is used to forecast safety stocks over the forecast horizon, and a periodic replenishment model is then used to generate unconstrained replenishment signals. Figure 1-5 depicts this process at a high level. Capacity constraints are then added, permitting the generation of constrained replenishment signals by working backwards from the forecast horizon.

Historical simulations of the model's performance on more than 75% of Intel's CPU production volume indicate that total finished goods realized through the use of this model would exceed 50%, with a net present value in excess of $80 million.

Figure 1-5. Inventory Model Process Flow


Historical Shipment Data
Generate Shipment Forecast
Day-of-Week Factors
Shipment Forecast Error Distribution
Historical Intel Forecast Data
Generate Forecast of Intel Forecast Error
Intel Forecast Error Distribution
Convolve Forecast Errors over Lead Time
Safety Stock Forecast
Generate Unconstrained Inventory Goals
Generate Unconstrained Daily Replenishments
Lead Time Forecast Error Distribution
Historical Lead Time Data
Generate Lead Time Forecast
Future Demand Forecast Data

1.4 OVERVIEW OF THESIS

In this chapter, we have overviewed the supply chain challenge faced by Intel and other semiconductor manufacturers. We have introduced motives for improved inventory management systems and also have outlined our proposed solution for the automated management of finished goods inventories.

In the next chapter, we present current forecasting methods and a review of relevant forecasting methods. A new form of exponential smoothing that forms the foundation of our inventory model is introduced. Chapter 3 reviews current inventory management methods and the base stock model used for generation of unconstrained and constrained replenishment signals. Results of historical simulations are also presented. Chapter 4 outlines the organizational impact of converting to an automated inventory replenishment system and offers recommendations for implementation and further improvements to the model. Finally, Chapter 5 presents conclusions and key insights for this thesis.

2 DEMAND FORECASTING METHODS AND MODELS

In this section we examine Intel's current forecasting methods and present a model to allow for their integration into an inventory management system. A review of existing mathematical approaches is presented, followed by a new form of exponential smoothing that addresses the problems faced by Intel that existing approaches cannot solve.

2.1 CURRENT INTEL FORECASTING METHODS

As with most corporations, demand forecasting at Intel is a complex process involving many people, data sources, and techniques. Intel's methods can best be described as a combination of time series and causal techniques that are then subjected to qualitative judgments. While the use of consistent, un-manipulated mathematical techniques for generating forecasts generally is considered to be a method superior to human judgment and manipulation [Chambers, Mullick, and Smith, 1971], it is uncommon in practice and nearly impossible to put into place. Thus, we make no attempt to improve Intel's current forecasting techniques beyond the addition of a new technique that can be layered on top of current methods in order to facilitate more rigorous inventory management.

The form of the forecast utilized by our model varies depending on the forecast horizon. Short-term forecasts are point estimates of weekly demand at the line-item level. Medium-term forecasts are also point estimates at the line-item level but are usually monthly quantities rather than weekly. In general, we only need forecast data over the projected lead-time plus any additional time required to pre-build inventory in anticipation of a capacity constraint. Lead-times range from as few as three days to as many as two months, depending on the production stage and product type. As such, we are generally concerned with short-term forecasts. Medium-term forecast data are only sometimes required, and long-range forecasts are not relevant to short-term inventory control.

2.2 Existing Mathematical Approaches

The development of any forecasting system is intimately tied to an understanding of the salient features of the demand data. Observations of seasonality, trends, correlation, and other factors must be taken into account in order to choose the correct forecast method. In Intel's case, we find that the structure of demand for CPU products is markedly different from that of Flash Memory. Many CPU line items exhibit 7-day seasonal patterns, whereas Flash Memory products do not. In addition, the demand characteristics of line items within each product type vary widely depending on the maturity and popularity of the product.

Figures 2-1 through 2-4 present demand for four representative CPU products. The full dataset is presented in Appendix A. The data in these graphs have been normalized such that the mean demand over the model-training period is 100 units. As the graphs depict, demand volatility is quite high. CPU Line Item 1 depicts nearly an entire lifecycle for a typical product. Line Item 2 is similar, but exhibits stronger day-of-week seasonal characteristics. Line Item 3 tests the model against a product that is ramping up, whereas Line Item 4 tests the model against a product with a relatively stationary demand process.

Figure 2-1. CPU Line Item 1 Demand

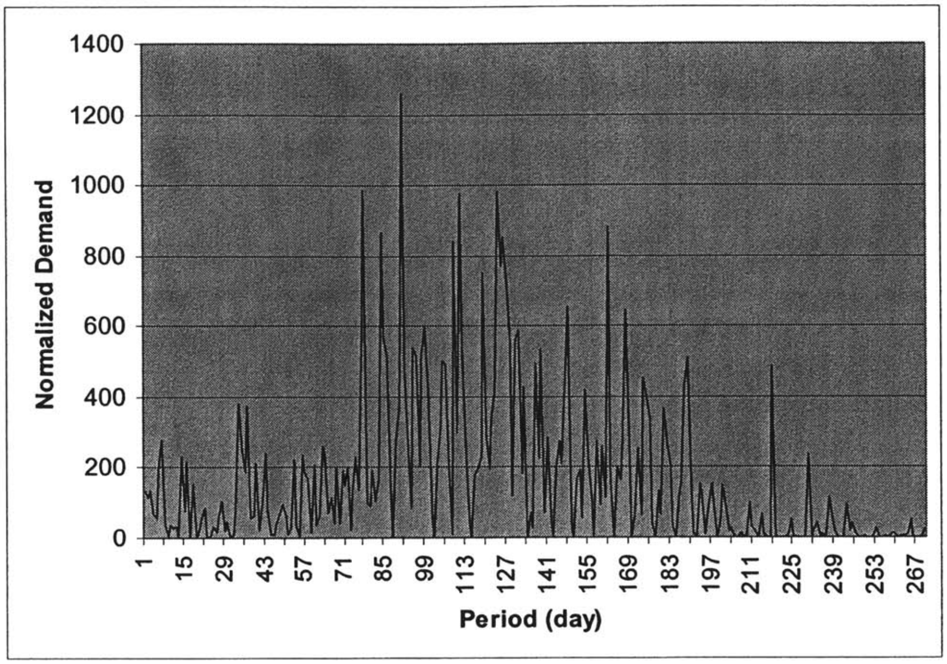


Figure 2-2. CPU Line Item 2 Demand

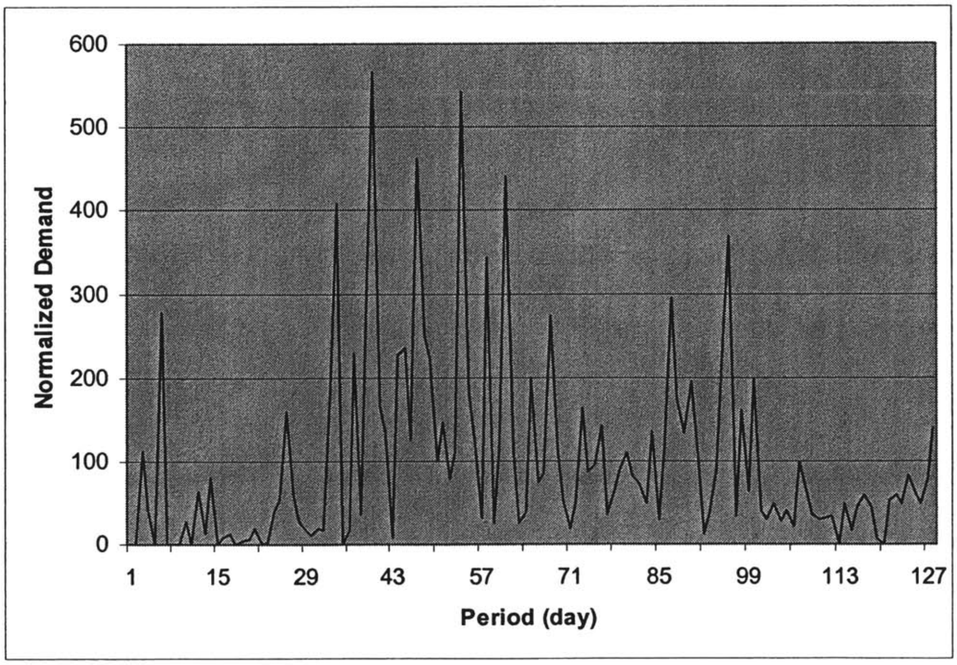


Figure 2-3. CPU Line Item 3 Demand

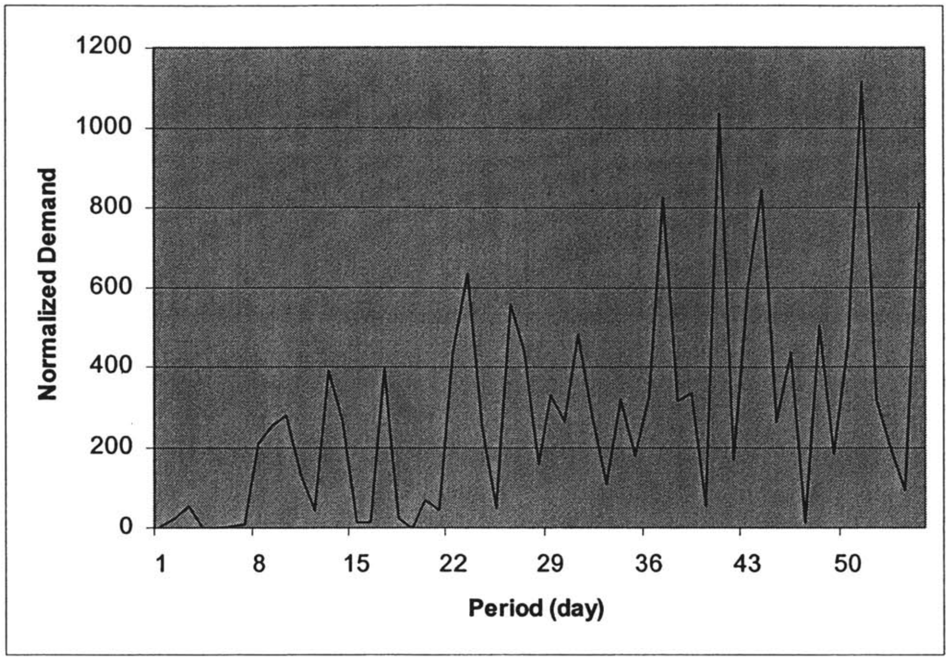
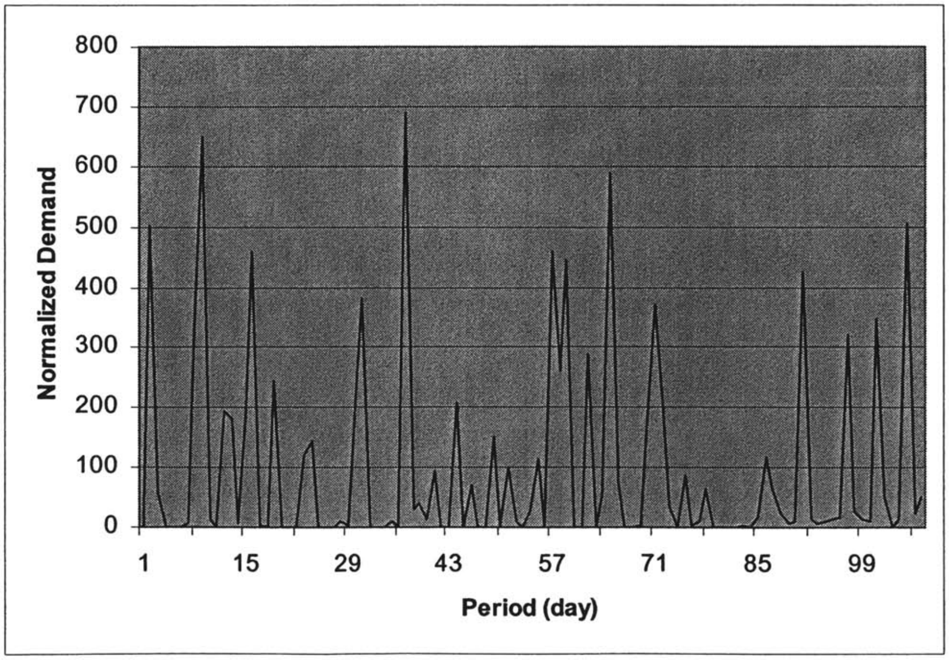


Figure 2-4. CPU Line Item 4 Demand



2.2.1 Overview of Forecasting Techniques

The high clockspeed[3] of the industry, relatively short product lifecycles, and complex characteristics of demand quickly narrow our choice of forecasting techniques. The key factors in choosing our forecasting technique are that we are estimating daily demand over the short-term and that we wish to incorporate day-of-the-week seasonal effects. Causal models such as econometric techniques based on price elasticities are capable of these types of forecasts but are not capable of estimating weekly seasonal effects. Qualitative methods could be used, but manual estimation of each day's demand would be extremely labor intensive and would provide relatively poor forecasts. In addition, their adaptation to mathematical inventory control would require huge quantities of historical forecast data. Thankfully, time-series methods are well suited to our forecasting problem.

Time-series analyses include moving averages, exponential smoothing, Box-Jenkins, and other ARIMA methods [Borchers, 2001], trend projection, and more [Chambers et al, 1971]. We have chosen a combination of moving average and exponential smoothing techniques due to their relative simplicity, proven robustness against various types of demand patterns, and popularity in solving forecasting and inventory control problems [Gardner, 1985].

2.2.2 Exponential Smoothing Models

Exponential smoothing methods come in many variations, each capable of handling different degrees of trends and seasonality. All of these methods use a weighted

[3] Clockspeed is a term used to describe the rate of change faced by the firm in the form of product, process, or organizational innovation [Fine, 1998].

average of the most recent demand and the previous forecast. The error-correcting form of simple exponential smoothing is as follows:

$$Level_t = Level_{t-1} + \alpha * Error_t$$

where $Error_t = Actual_t - Level_{t-1}$ and the most recent value of *Level* represents the forecast for all future periods.

Two popular schemes that include trend and seasonal terms are the Holt-Winters model [Holt et al., 1960, Winters, 1960] and the Brown model [Brown, 1963]. Holt-Winters is more flexible and accurate but more difficult to optimize due to its use of three smoothing parameters. The error-correcting form of the Holt-Winters method in its unaltered state is as follows [Gardner, 1985]:

$$Level_t = Level_{t-1} + Trend_{t-1} + {}^{\alpha * Error_t}\!\big/_{Seasonal_{t-p}}$$

$$Trend_t = Trend_{t-1} + {}^{\alpha * \gamma * Error_t}\!\big/_{Seasonal_{t-p}}$$

$$Seasonal_t = Seasonal_{t-p} + {}^{\delta * (1-\alpha) * Error_t}\!\big/_{Level_t}$$

$$Error_t = \left(Actual_t - \left(Level_{t-1} + Trend_{t-1}\right) * Seasonal_{t-p}\right) * X_t$$

$$X_t = \begin{cases} 0 & \text{if } Actual_t = 0 \\ 1 & \text{otherwise} \end{cases}$$

$$Forecast_t(k) = \left(Level_t + k * Trend_t\right) * Seasonal_{t-p+k} \quad \text{w.p.} \quad \frac{\left(\sum_{i=0}^{t} X_i\right)}{t}$$

where *p* represents the length of the season and *k* is the number of periods into the future for which we desire a forecast.

The Brown model is structurally similar and we therefore do not present it here. As in the simple exponential smoothing model, the forecasted level is constant for all future periods. However, the forecast is modified by the trend and seasonal factors.

There are three major shortcomings of these models with respect to our forecasting problem:

- Only positive demand days are counted.
- The lag on seasonal factors is dependent on the level smoothing coefficient (α).
- The error vector is constant across all levels.

Each of these facts requires careful treatment and has important bearing on the development and application of our model.

2.2.3 Handling Zero Demand

Traditionally, there are two approaches to handling zero demand: either treat no observed demand as a zero, or ignore observations of zero in the model. Each has its advantages and disadvantages, but in our case the disadvantages of ignoring observations of zero eliminate it as a possible method for solving our forecasting problem.

The inclusion of observations of zero demand eliminates the need for estimating the probability of demand occurring (the Bernoulli factor X_t described in the Holt-Winters model). However, a model such as this breaks down when confronted with low-velocity demand patterns because a high frequency of zero demand skews the error distribution and increases the variance of the error vector. In some cases, low-velocity demand can cause the level to drop so much that it introduces large errors when combined with multiplicative seasonal factors. This problem is most obvious when one of the days within the season is always zero – the model essentially breaks down in this case.

The elimination of observations of zero demand also presents problems. Because we require seasonal factors for each day of the week, we must track a separate Bernoulli factor for each day of the week in order to eliminate confusion between Bernoulli effects and seasonality. We can imagine a case where the probability of demand occurring on a Saturday or Sunday is very low, but the probability of demand occurring on a weekday is nearly 100 percent. If we were to use a single estimate for the probability of positive demand across all days of the week, the seasonal factors for weekdays would be incorrectly skewed downward by the weekend effect. We can track separate effects, but in order for these values to be statistically significant, we need almost the entire product history for a line item!

Most importantly, it is necessary to recall that for our case we must combine our model with Intel's existing weekly forecasts in order to produce a daily forecast. The correct estimation of daily demand in the future cannot be accomplished merely by multiplying the appropriate Bernoulli effect into the demand forecast. Monte Carlo simulation is required in order to properly estimate the full distribution of demand on a given day within the forecast horizon. Such a simulation would require 500 to 1000 trials for each day's demand forecast – this is not practical when confronted with the need to do this every week for hundreds to thousands of line items.

The elimination of zero demand observations has too many disadvantages to be a viable choice of method for our model. The inclusion of these observations simplifies the model but limits its application to demand patterns where the percentage of zero demand days is lower than 25%. Thankfully, this is not a major problem for Intel, as the vast majority of its demand volume falls into this category.

2.2.4 Mismatch Between Historical and Forecasted Levels

The use of exponential smoothing on the level produces a mismatch between the seasonal factors and the level when we try to pair it with Intel's weekly forecasts. With the Holt-Winters and Brown methods, the forecast simply combines the last level with the trend and the seasonal factors. The seasonal factors have been trained[4] using the historical data and the level generated by the level smoothing coefficient. This coefficient produces a lag on the responsiveness of the level to the data that is inversely proportional to the coefficient. In other words, a coefficient of 0.1 is indicative of a level that corresponds to a lag of ten data points - the weighted average "age" of the data points is about ten. As the lag increases, the responsiveness of the level to recent data points decreases.

The problem arises when we seek to combine seasonal factors trained on a level generated by exponential smoothing of daily historical values with future level estimates derived from Intel's weekly forecasts. Because the seasonal factors are trained in tandem with the level, modifications to the responsiveness of the level will affect the seasonal factors. Future level estimates must therefore be structurally similar – i.e. of similar responsiveness – in order to minimize the error of our daily forecasts when the seasonal factors are combined with the level.

[4] For the purposes of this thesis, model training refers to the minimization of the mean squared error generated by running the model through a set of historical data. Optimization is used to choose smoothing coefficients and the "trained" seasonal factors are the most recent set of factors generated by the model from the historical data.

We cannot simply apply the smoothing coefficient generated from daily values to exponential smoothing of weekly forecasts without producing a lag mismatch. The use of a moving average to generate a daily forecast from Intel's weekly forecasts overcomes this problem. While we could then use exponential smoothing on this daily forecast, we are again confronted with a lag mismatch, as the moving average has its own lag of length equal to the number of data points in the moving average.

The mismatch in the types of levels and their lags will lead to errors in the forecast. While some of this error could be reduced by altering the length of the moving average for the forecasted level to the inverse of the historical level smoothing coefficient, we cannot avoid the underlying mismatch in the structure of the levels. In addition, the optimal choice of smoothing coefficient may be very low, resulting in a moving average with a length that exceeds the forecast horizon!

Our solution to this problem is to replace exponential smoothing for the historical level with a moving average of length equal to that used for generation of the forecasted level.

2.2.5 Drawbacks of Non-Multiplicative Error

The use of an error distribution that is constant across all levels results in inflated inventories for products with highly non-stationary demand levels. Most, if not all, of the exponential smoothing methods commonly used for forecasting, including the Holt-Winters and Brown methods, generate an error vector that is constant across all levels. For slowly changing levels, this does not present a large problem. However, for short lifecycle products such as those produced by Intel, or for products that are undergoing large changes in the average level of demand, a constant error distribution ignores the possibility of positive correlation between the volatility of demand and the mean of demand. While it is possible that this correlation might be zero, we have found that this is almost never the case – as demand rises, so does the volatility of that demand. Indeed, most corporations implicitly make this assumption through the use of inventory policies that set inventory targets in days or weeks of demand.

To illustrate this shortcoming, imagine that our lead time is 60 days. Point forecasts of future demand indicate that weekly demand over the next 60 days will double, on average. If we were to base our safety stock estimates on an error vector that is not proportional to the average demand level, our safety stock target would remain unchanged despite the increase in demand. If the volatility of demand did increase, our stock out risk would rise as well!

Therefore, our chosen forecasting methodology must allow error to be proportional to some estimate of the average level of demand. While the degree of correlation will vary from line item to line item, we have found that it is closer to +1 than 0 in most cases. One could imagine a model that allowed flexibility in the degree of proportionality through a fractional exponent based on the degree of correlation; however, it is our belief that this would complicate the model for minimal gain in inventory reduction. As a result, we have chosen a model where error is directly proportional to demand level.

2.2.6 Motivation for New Model

We believe that we have shown that exponential smoothing methods offer the best possibility for estimating demand over a short-term forecast horizon. The Holt-Winters and Brown methods provide a reasonable starting point for developing a seasonal model but cannot solve our problem in their pure form due to their treatment of observations of zero demand, inability to be combined with other forecasts, and constant error distributions. In addition, their application to our data sets yields unstable solutions for smoothing coefficients and trend coefficients that are too high to provide useful forecasts.

In the following section, we propose a new method of exponential smoothing that overcomes the failings of traditional exponential smoothing models.

2.3 A NEW FORECASTING MODEL

Our model is best described as a variation of exponential smoothing in which seasonal factors are trained on the ratio between actual values and the level estimate. The level is produced using a seven-day moving average rather than an exponentially weighted moving average, and trend techniques are not employed. Optimization techniques are used to choose the seasonal smoothing coefficient that minimizes the squared seasonal error. The final set of seasonal factors is paired with weekly point forecasts of demand (generated through exogenous means not examined by this thesis) in order to estimate daily demand and error over the forecast horizon. A detailed mathematical description of this model follows.

2.3.1 Multiplicative Forecast Error Model

The conversion from gross demand quantities from Intel's forecasts to a daily forecasted level is determined as follows. A weekly point forecast is used in this case; monthly or quarterly quantities could be converted in similar fashion:

$$ForecastLevel_t = \begin{cases} \dfrac{\sum_{i=t-3}^{n} Actual_i + (y+3) * WeeklyForecast_{ThisWeek} / 7}{7} & \text{for t = n+1 to n+3} \\ \dfrac{(4-y) * WeeklyForecast_{LastWeek} + (y+3) * WeeklyForecast_{ThisWeek}}{49} & \text{for t > n+3, } y \leq 4 \\ \dfrac{(11-y) * WeeklyForecast_{ThisWeek} + (y-4) * WeeklyForecast_{NextWeek}}{49} & \text{for t > n+3, } y > 4 \end{cases}$$

where $y = remainder\left(t/7\right)$ and *n* is the number of historical data points. Thus, the forecasted daily level is a seven-day moving average centered on the day in question (i.e. no lag). Due to the use of the moving average with no lag a combination of forecasted demand and historical values must be used if t ≤ n+3. We have chosen this form for our forecasted level in order to match the structure of the historical level.

The level is determined as follows. Note that because we are concerned solely with training the model to produce seasonal factors and error estimates that will best predict the future (as opposed to minimizing historical error), the level is calculated with actual values that are centered on the time period in question, rather than lagging or leading:

$$Level_t = \begin{cases} Level_{t+7} & \text{for t = 1 to 3} \\ \dfrac{\sum_{i=t-3}^{t+3} Actual_i}{7} & \text{for t = 4 to n-4} \\ \dfrac{\sum_{i=t}^{n} Actual_i + \sum_{j=n+1}^{t+6} ForecastLevel_j}{7} & \text{for t = n-3 to n} \end{cases}$$

Thus, the historical level is a seven-day moving average of historical values with the use of the forecasted level for t ≥ n-3.

The remainder of the forecasting model is as follows:

$$ActualRatio_t = \begin{cases} \dfrac{Actual_t}{Level_t} \text{ for } Level_t > 0 \\ Seasonal_{t-7} \text{ otherwise} \end{cases}$$

$$SeasonalError_t = ActualRatio_t - Seasonal_{t-7}$$

$$Seasonal_t = \begin{cases} 1 \text{ if } \delta = 0 \\ \text{otherwise} \\ \quad Seasonal_{t-7} + \delta * SeasonalError_t \text{ for t > 0} \\ \quad \dfrac{7 * Actual_t}{\sum_{i=-6}^{0} Actual_i} \text{ for t = -6 to 0} \end{cases}$$

where δ is the seasonal smoothing coefficient and the values of *Seasonal* are the seasonal factors.

Notice that when the level is zero, the seasonal error is also zero. Additionally, the seasonality component collapses if the seasonal smoothing coefficient is zero. We have also chosen to set initial values for the seasonal factors using only the first week of data from the training set. When the entire training set is used, the result is inevitably that the model is overfit and the seasonal smoothing coefficient is set to extremely low values.

The model is trained by running the data forwards, backwards, and then forwards again. Godfrey and Powell (2000) have shown that more than a single cycle of forward and backward passes tends to overfit the data, resulting in less accurate forecasts. However, due to the use of seasonality, the initial forward pass must be for the maximum number of periods that is a multiple of the length of the season. The number of periods is thus defined as $n = 2m + l$. In addition, the backwards pass must run the seasons backwards, but not the periods within each season.

The seasonal smoothing coefficient, δ, is chosen using an optimization algorithm to minimize the following criterion:

$$\sum_{i=1}^{n} (SeasonalError_i)^2 \text{ such that } 0 \le \delta \le 1.$$

Forecast for k periods into the future are then generated as follows:

$$Forecast(n+k) = Seasonal_{n-q} * ForecastLevel_{n+k}$$

where $q = 7 - remainder\left(\frac{k}{7}\right)$.

2.3.2 A Note on Model Training[4]

The training of time-series models presents problems to those seeking to use them on short lifecycle products, because time is something that is in short supply. A model that can only be used for half of a product's lifecycle is not of much use in the business world. As such, we must determine the length of data set required to train the model as well as find ways to shorten it as much as possible.

Tests of our model have shown it to produce good results with as little as four seasons (weeks) worth of training data, although seven weeks' or more is recommended. In Intel's case, we find that this does not place undue strain on the use of the model due to the generational nature of Intel's products at the line item level.

A training set is required for the first generation of a particular product. However, as the product is transitioned from one stepping to the next, the underlying demand transitions as well. The implication of this is that we may jump start subsequent generations using the seasonal factors and seasonal smoothing coefficient from the previous generation. After sufficient time has passed to build up a set of data on the new generation, the model can be re-initialized and re-optimized.

We have not examined the uniqueness of model solutions. In general, we have found that optimization results converge consistently, regardless of the seed value used for the seasonal smoothing coefficient. To speed optimization, we recommend using a seed value between 0.15 and 0.25.

Additionally, optimization is not required every time a forecast is generated. The optimal smoothing coefficient does not shift appreciably from week to week. We therefore recommend re-optimization on a monthly basis, unless obvious structural changes occur in the demand pattern (such as a sudden shift in demand characteristics due to the loss of a major customer).

2.4 FORECASTING RESULTS AND CONCLUSIONS

We have chosen to compare our model with two other methods: simple exponential smoothing and the Holt-Winters method with trend removed. Tables 2-1 through 2-3 present the results of this comparison. In these tables, MFE refers to the multiplicative forecasting error method we have presented, SES refers to simple exponential smoothing, and HWS refers to the Holt-Winters smoothing method.

Initial values for seasonal factors for HWS are set in the same manner as described for MFE in Section 2.3.1. Initial values for the level are equal to the actual value for t=0 for both SES and HWS. Optimizations are completed using the "Solver Add-in" for Microsoft Excel® [5].

Table 2-1. Training Period Comparisons

	Training Period (weeks) **		
Item	MFE	SES	HWS
1	7+1	8	7+1
2	7+1	8	7+1
3	3+1	4	3+1
4	7+1	8	7+1

** +1 indicates number of weeks used for training seasonal factors but not included in training passes

Table 2-2 Mean Square Error Comparisons

Item	Optimization Method	Forecasting Method	α	δ	Training MSE	Testing MSE
1	Training Set	MFE	N/A	0.0445	5814	22081
	Training Set	SES	0.214	N/A	9604	41640
	Training Set	HWS	0.0825	0.0399	8973	31399
1	Testing Set	MFE	N/A	0.0445	5814	22081
	Testing Set	SES	0.115	N/A	9798	40759
	Testing Set	HWS	0.108	0.175	9651	30637
2	Training Set	MFE	N/A	0.305	7718	4644
	Training Set	SES	0.168	N/A	12648	9260
	Training Set	HWS	0.000194	0.659	12803	7724
2	Testing Set	MFE	N/A	0.332	7547	4552
	Testing Set	SES	0.075	N/A	13575	8985
	Testing Set	HWS	0.158	0.367	74273157	7022
3	Training Set	MFE	N/A	0.290	14172	79596
	Training Set	SES	0.000	N/A	14845	168352
	Training Set	HWS	0.001	0.159	16957	98974
3	Testing Set	MFE	N/A	0	15360	85039
	Testing Set	SES	0.121	N/A	15540	90776
	Testing Set	HWS	0.030	0.391	19722	58769
4	Training Set	MFE	N/A	0.351	18633	24565
	Training Set	SES	0.029	N/A	31904	23482
	Training Set	HWS	0	0.368	20164	25106
4	Testing Set	MFE	N/A	0.230	19976	23009
	Testing Set	SES	0.010	N/A	32696	23193
	Testing Set	HWS	0	0.269	20467	24608

[5] Excel is a registered trademark of Microsoft Corporation.

In Table 2-2 we present two sets of optimization results for each line item – only the objective function differs. For sets labeled "Training Set," the objective is to minimize mean squared error (MSE) during the training period, whereas for sets labeled "Testing Set" the objective uses the MSE during the testing set. We have done this to provide a lower bound for error during the testing period that can be compared to values generated with the training set.

We find that MFE provides the lowest mean square error during the training period in all cases. More importantly, MFE also provides the lowest error during the testing period for all of the training sets except for CPU Line Item 4. This is the true test of the model, since a model optimized using only training data most closely resembles a real application of the model. In the case of CPU Line Item 4, all three methods produce similar results, which is to be expected given the more stationary nature of the demand process.

Table 2-3. Seasonal Factors at End of Training Period

		Trained Seasonal Factors	
Item	Day of Season	MFE	HWS
1	n+1	1.336	1.161
	n+2	0.905	0.841
	n+3	0.473	0.627
	n+4	0.757	0.664
	n+5	0.660	0.644
	n+6	0.942	0.905
	n+7	1.811	1.602
2	n+1	0.092	0.365
	n+2	0.669	6.675
	n+3	0.815	5.462
	n+4	0.655	4.818
	n+5	2.084	14.807
	n+6	1.555	7.740
	n+7	0.897	5.846
3	n+1	0.674	1.007
	n+2	0.904	1.008
	n+3	2.655	2.641
	n+4	0.330	0.846
	n+5	0.067	0.168
	n+6	1.082	2.247
	n+7	0.701	1.195
4	n+1	0.221	0.764
	n+2	4.160	3.111
	n+3	1.113	1.095
	n+4	0.483	0.328
	n+5	0.250	0.713
	n+6	0.459	0.816
	n+7	1.141	0.529

In all four data sets, we find that demand exhibits strong seasonality. The seasonal factors for MFE and HWS differ due to the different methods used for calculating the level the factors refer to. In the case of CPU Line Item 2, note how the HWS model produced extreme values for the seasonal factors. This is the result of a low value for the level smoothing coefficient combined with a high value for the seasonal smoothing coefficient. When demand shifts upwards, the seasonal factors must shift dramatically upward to account for the lag on the level. Additionally, there is a dramatic difference in the optimal smoothing coefficient for the two sets of optimizations for CPU Line Item 2. In sum, for HWS the level is highly dependent on the data set used to optimize the smoothing coefficients, implying that this method may pose difficulties in practice.

The smoothing coefficients for MFE and SES also shift depending on the optimization method used. However, the mean squared error does not change dramatically for these methods, unlike HWS, implying that they are more robust to structural changes in the demand process. However, SES produces higher mean squared error in most cases.

The implication of these results is that our method may be superior to both simple exponential smoothing and the Holt-Winters method for products with non-stationary demand processes with seasonality. Additionally, the lack of large shifts in error despite changes in the smoothing coefficient suggests that frequent re-optimization is unnecessary.

3 INTEGRATION WITH INVENTORY MANAGEMENT

In this section we integrate the forecasting method presented in Section 2 with a model for inventory control. A review of current methods for inventory management at Intel is presented, along with a discussion of non-stationary inventory policies and the base stock model. Methods for convolving lead time demand are discussed, followed by the presentation of our integrated model for both unconstrained and constrained replenishment policies. Finally, the results of our model are compared to current methods, revealing dramatic inventory reduction possibilities.

3.1 CURRENT INVENTORY MANAGEMENT METHODS

Traditionally, Intel Corporation has managed its major inventory points through experience and judgment alone. Inventory goals are typically expressed in weeks of demand, using a form of moving average based on a quarterly time frame. We seek to provide methods to allow Intel to accomplish a paradigm shift in their supply chain management practices by converting from a push system to a pseudo-pull system.

A push system is traditionally defined as one in which production decisions are based solely on demand forecasts without regard to historical estimates of the distribution of demand, lead time, and other factors. Conversely, under a pull system these historical estimates are employed to derive inventory goals that are then compared to on hand inventories to determine production decisions. A traditional pull system does not employ forecasts. We classify our model as a psuedo-pull system because while it makes use of Intel's forecasts, it also uses historical estimates of the distribution of demand, lead times, and Intel's forecast error to determine forecasts of inventory goals, which in turn drive production quantities. Production quantities are “pulled through” to finished goods inventory, rather than pushed into inventory to meet demand forecasts.

A cursory examination of the challenge Intel’s supply chain managers face quickly reveals why statistical methods have not yet been employed:

- Six major inventory points, dozens of warehouses and factories, two major manufacturing stages, tens of thousands of finished line items, and hundreds of customers make for a complex supply network, rather than a simple supply chain.
- Product lifecycles for most products are less than 18 months, with the majority of demand occurring in a 3 to 6 month timeframe.
- Production yields are highly stochastic.
- Manufacturing technology is continually changing and among the most sophisticated in the world.
- Lead times from raw materials to finished goods are extremely long (months).
- Customers and Intel factories are spread worldwide.
- Demand and lead times are highly stochastic, with non-stationary means and probability distributions.
- Manufacturing equipment is shared across a diversity of product types (Flash Memory, CPU, Communications Products, etc.), each with a different set of customers and market conditions.

We turn our attention next to the selection of a model that best fits the supply chain challenge faced by Intel Corporation.

3.2 Handling Non-Stationary Demand Distributions

Most inventory models used by academics and corporations today rely on an assumption of stationary demand distributions. On the surface this seems to be at odds with the non-stationary nature of many forecasting methods such as exponential smoothing. Indeed, we already have argued that the assumption of stationary demand processes in the future is unwise for products that have shown non-stationary tendencies in the past. However, this does not mean that traditional inventory models cannot be applied.

One key to the successful use of stationary inventory models is to find a forecasting methodology that produces an error distribution that remains stationary despite the non-stationary nature of the underlying demand process. Our model does exactly this by generating an error distribution that remains a fixed in proportion to the forecasted level. This allows us to convolve a single estimate across all future periods of what percentage of the forecasted level should be carried as safety stock. This estimate can then be used to forecast evolving inventory goals.

3.3 The Inventory Model

Base Stock Theory forms the underpinning of our inventory model. A detailed explanation of this type of model can be found in Silver and Peterson (1985). We have chosen to use a periodic, order-up-to model due to the periodic nature of Intel's existing internal planning processes as well as the likelihood that many of Intel's customers use periodic inventory replenishment policies.

We model inventories at a single stage. The expansion of this model to multiple stages is not examined, although we believe it could form the basis of a wider model.

Axsäter and Rosling (1993) present a detailed examination of multilevel inventory control models that could serve as a good starting point for this analysis. The optimality of our inventory policy is also not examined – while base stocks are optimal in certain classes of stationary problems, they are not necessarily optimal here. However, we expect them to work reasonably well here and in settings similar to those presented in this thesis. We begin with two methods for convolving lead time demand, followed by a presentation of the inventory model without and with capacity constraints.

3.3.1 Methods for Convolving Lead time Demand

The standard model for an order-up-to base stock policy is comprised of two types of inventory: pipeline inventory and safety stock. Pipeline inventory is determined through estimates of the average expected demand level over the lead time. However, the determination of safety stock requires more careful treatment due to its dependence on the probability distribution of expected demand over the lead time.

In Section 2, we presented a new model for forecasting demand. However, the seasonal error vector from this model only represents one of the ingredients necessary for the proper determination of safety stock targets. Because we have based our forecasting model on Intel's forecasts, we must include the error from these forecasts in our convolution of the probability distribution of lead time demand. In addition, in Intel's case lead times are stochastic and non-stationary and therefore must also be included.[6]

[6] We have chosen to neglect yield forecast error. Yields are also stochastic and non-stationary in Intel's case but they quickly reach predictable levels, and we therefore assume they can be factored into replenishment signals directly.

3.3.1.1 Direct Convolution

For the case of Intel's CPU products, lead times through Assembly/Test generally exceed 10 days. Following the Central Limit Theorem, we assume normal distributions for any daily errors extrapolated to ten or more days. For lead times less than 10 days, the discrete distribution methods presented in the next section may need to be employed.

We begin with the standard form for the interaction of demand and lead time variability:

$$SafetyStock = z^{-1}(ServiceLevel) * \sqrt{(ALT + RT) * \sigma^2_{Demand} + \mu^2_{Demand} * \sigma^2_{ALT}}$$

where $z^{-1}(ServiceLevel)$ refers to the z-factor for the desired service level, ALT refers to the average lead time, RT is the review period, and σ and μ take on their usual forms of standard deviation and mean, respectively.

It is important to remember that we are deriving a multiplicative factor for daily safety stock estimates, rather than a constant estimate for all future periods. We assume that the error of an Intel forecast for one week is independent from the error of another. We also assume that seasonal forecast error is identically distributed and independent of Intel's forecast error, lead time, and review period. Finally, we assume that Intel's weekly forecasts are dependent on lead time inasmuch as the quantity of forecast error is dependent on the forecast horizon (in weeks).

First, we must convert Intel's weekly forecast error distributions into a single daily error distribution. This is accomplished as follows:

$$\sigma^2_{Daily\_IFE} = \left[\sum_{k=1}^{ALT+RT} \frac{\sigma^2_{Weekly\_IFE\_k}}{7 * IntelWeeklyForecast\_k} \right] \Big/ (ALT + RT)$$

where *IntelWeeklyForecast_k* and $\sigma^2_{Weekly\_IFE\_k}$ respectively refer to Intel's forecast for demand and the variability of Intel's forecast error for the week *k* falls in. Note that Intel's forecast error is now in percentage terms (i.e. days of demand).

Because the variation of demand relative to Intel's forecast is dependent on both the seasonal forecast error and Intel's weekly forecast error, we must convolve these two together over the lead time and the review period. Additionally, we may remove the μ^2_{Demand} term because we are convolving a safety stock estimate that scales with the mean of demand. All terms are in terms of a percentage of the mean, therefore in this case μ is literally one mean and μ^2 is simply one. The formula for our multiplicative safety factor is thus as follows:

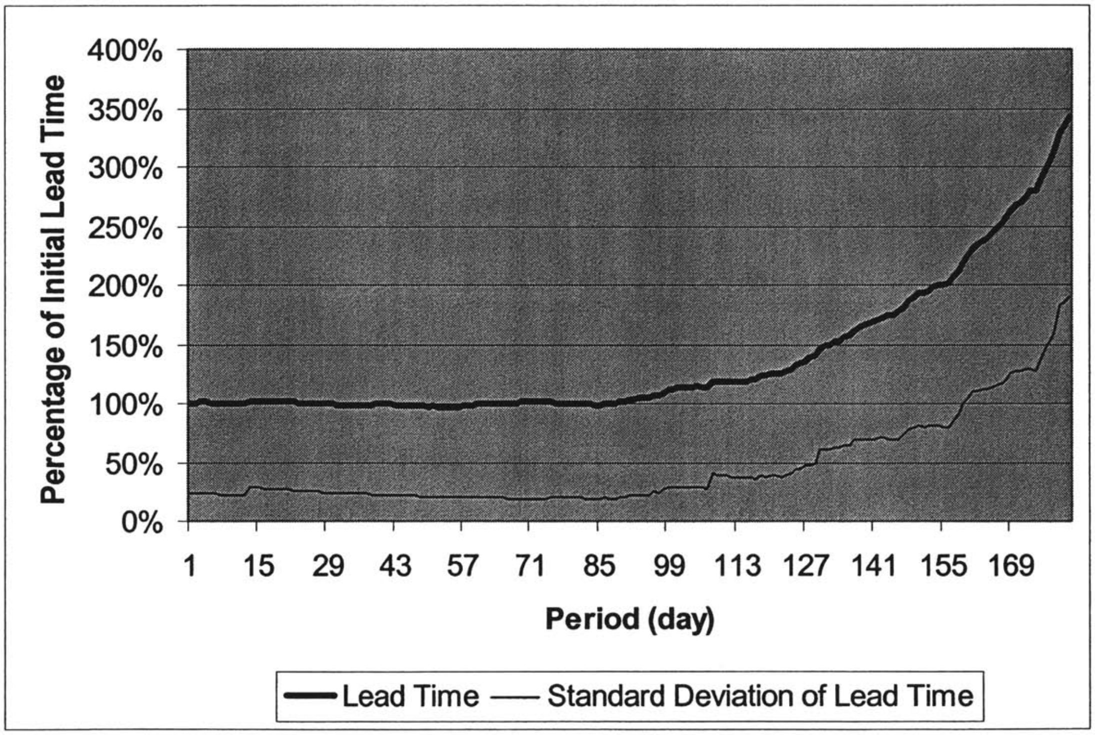
$$SafetyFactor = z^{-1}(ServiceLevel) * \sqrt{(ALT + RT) * \left(\sigma^2_{SeasonalError} + \sigma^2_{Daily\_IFE}\right) + \sigma^2_{LT}}$$

Although the lead time may be non-stationary, it is assumed to be stationary for the calculation of the required safety stock on a given day. However, the lead time used for one day's safety stock calculation may vary from that of another. In practice, we have found that so long as the lead time does not change too quickly from one period to the next, excessive bullwhip is not introduced into the system.

For the purposes of this thesis, we have assumed a lead time of twelve days, a review period of one day, and a lead time standard deviation of 20% of the nominal lead time (2.6 days) for each CPU line item. In similar fashion, we have assumed that forecast error for Intel's weekly demand forecasts is 5% for the first week out and increases by 2% for each week thereafter. These values are used for illustrative purposes only and are in no way representative of actual lead times or forecast errors experienced by Intel, which vary both over time and from product to product. Average lead times and forecast errors may in fact be much smaller or much larger than the values we have assumed.

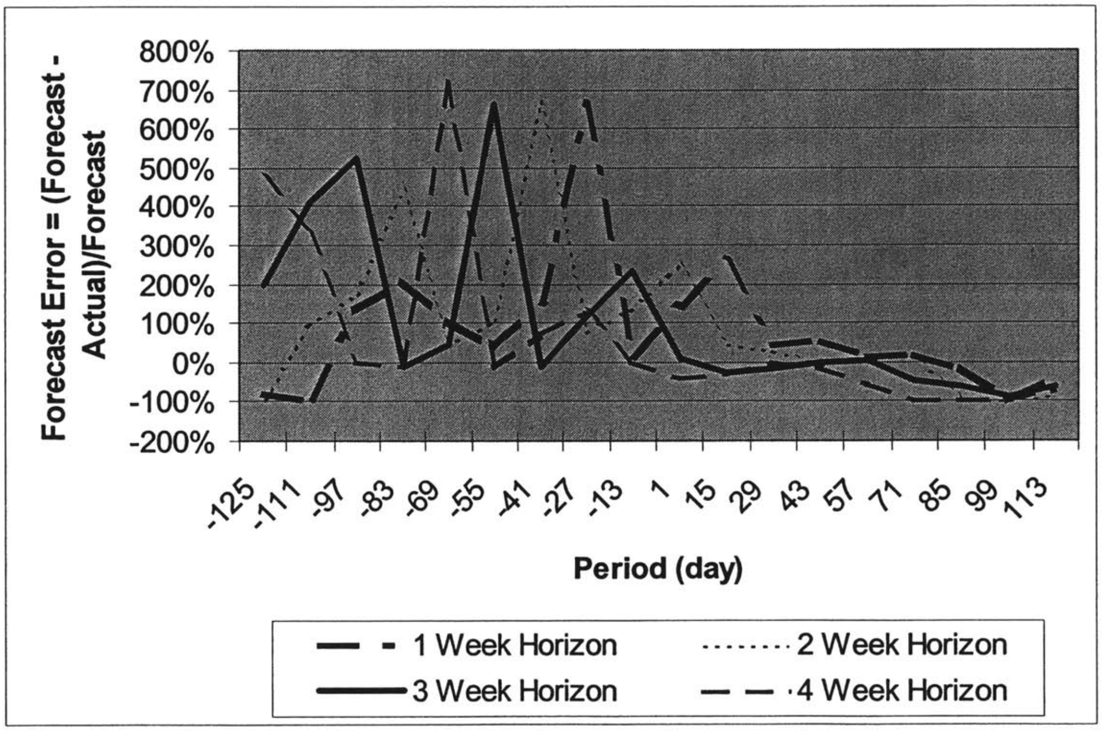
In actual fact, we find that for Intel, both lead time and lead time variability are highly non-stationary. Both values remain stable in the early parts of a product's lifecycle, then increase dramatically after a product has reached its peak demand rate, eventually reaching levels that make the use of a base stock model impossible. Figure 3-1 illustrates this behavior. Lead times and lead time variability were calculated using the adaptive variance exponential smoothing methods developed by Snyder [2002].

Figure 3-1. Lead Time Behavior for CPU Line Item 2



The form of Intel's forecasts presents an additional roadblock to the short-term implementation of our model. As described in Section 2.1, Intel utilizes a wide variety of personnel and forecasting techniques to make estimates of how much of its products to manufacture. However, amongst all of the types of forecasts available, from production starts to production completions and from promises to customers to available capacity, as far as we can determine, Intel does not generate a short- to medium-term forecast that describes estimates of what shipments will be required during a given time period (i.e. a week). The closest Intel forecast available is a one of production completions – the error of this forecast compared to actual shipments is shown in Figure 3-2. As can be seen from the figure, the degree of error present makes this forecast difficult to use for our purposes.

Figure 3-2. Production Forecast Error for CPU Line Item 2



Forecast error in Figure 3-2 is calculated in the traditional simple fashion. However, it is highly likely that correlation in forecast error exists across generations, line items within a product line, and product types. Çakanyildirim and Roundy [2002] present an excellent method for estimating correlation and estimates of future forecast error for individual products. However, their methods require vast amounts of data (five years or more) and have proved infeasible for our purposes.

An examination of performance drivers for planning and manufacturing and close scrutiny of week-to-week changes in requests for production revealed many of the reasons for the increases in lead times. A detailed examination of these problems is beyond the scope of this thesis. However, suggested recommendations for solutions that better fit the implementation of this model are presented in Section 4. We next examine convolution through simulation.

3.3.1.2 Convolution through Simulation

For short lead time products (less than 10 days) with unusual probability distributions, the errors of the multiple time periods have to be analyzed by convolution. This is accomplished through the simulation of discrete probability distributions of lead time, seasonal error, and Intel's forecast error. Simulation methods produce good results and are more robust to unusual probability distributions, but require substantially more processing power.

Murty [2000] describes a form of this technique where the error distribution from an exponential smoothing of demand is split into discrete elements and convolved over the lead time using Monte Carlo simulation. Monte Carlo techniques operate through random draws on each probability distribution that are then processed to form a distribution of simulated results. A minimum of 500 to 1000 trials with 15 to 25 discrete elements is recommended for reliable results. Crystal Ball™ [7] is an excellent tool for the application of these techniques.

[7] Crystal Ball is a registered trademark of Decisioneering, Inc.

Either direct convolution or simulation may be used to produce an estimate for the required safety factor, or number of days' safety stock, that is required to meet a desired service level. The remainder of the inventory model is presented next.

3.3.2 Unconstrained Inventory Model

In manufacturing, there are two major types of buffers against uncertainty: production capacity and inventory. Choosing an optimal balance between the two is a difficult task and has been addressed by many capable authors in the literature. We find Atkinson's solution [2001] to be the most elegant and complete. These techniques require the extensive use of financial considerations that are beyond the scope of our model. However, we believe that our method could form the basis of numerous types financial optimization models.

We choose to address the requests for inventory replenishment when capacity constraints for a single product are known across the entire forecast horizon. The calculation of constrained replenishment forecasts first requires the determination of unconstrained replenishment signals. The unconstrained model is as follows:

$$BaseStockGoal_t = \sum_{i=1}^{ALT+RT} Forecast_{t+i} + Forecast_{t+ALT+RT} * (SafetyFactor)$$

$$CurrentStock_t = \sum_{i=1}^{ALT} replenishment_{t-i} + EndingOnHand_t$$

$$EndingOnHand_t = BeginningOnHand_t - Forecast_t$$

$$BeginningOnHand_t = EndingOnHand_{t-1} + replenishment_{t-(ALT+1)}$$

$$replenishment_t = \begin{cases} BaseStockGoal_t - CurrentStock_t & \text{if greater than zero} \\ 0 & \text{otherwise} \end{cases}$$

Note that this will yield current stocks equal to the safety stock for t > n+ALT+RT.

3.3.3 Constrained Inventory Model

In the semiconductor industry inventory is utilized far more often than capacity as a buffer due to the expensive nature of capacity. In addition, we have shown in Table 2-3 that daily demand seasonality is extreme in many cases. The leveling of this seasonality is accomplished through constraints on capacity. The optimal allocation of capacity between different products is dependent on many factors - our model does not seek to accomplish any such optimization. We seek to describe only the required replenishments to fit a given set of capacity constraints over the forecast horizon. The constrained model is as follows:

$$Shortfall_t = \begin{cases} replenishment_t - Capacity_t & \text{if greater than zero} \\ 0 \text{ otherwise} \end{cases}$$

$$prebuild_t = \begin{cases} Shortfall_t \text{ if } prebuild_{t+1} = 0 \\ \text{otherwise if } Shortfall_t > 0 \\ \quad Shortfall_t + prebuild_{t+1} \\ \quad \text{otherwise if } prebuild_{t+1} - (Capacity_t - replenishment_t) > 0 \\ \qquad prebuild_{t+1} - (Capacity_t - replenishment_t) \\ \qquad 0 \text{ otherwise} \end{cases}$$

$$Constrained\_replenishment = \begin{cases} Capacity_t & \text{if } prebuild_t > 0 \\ prebuild_{t+1} + replenishment_t & \text{otherwise} \end{cases}$$

This model works by comparing unconstrained replenishment signals to available capacities, working backwards from the forecast horizon. If capacity is unavailable to meet the required replenishment, then the replenishment must be ordered in earlier periods. The advantage of a model such as this is that allows for linear optimization techniques to be employed in the allocation of total factory capacity across Intel's many products.

3.4 Historical Simulation Results

We compare the results of our inventory model for CPU Line Items 1 through 4 to actual inventories held by Intel over the same periods. Actual inventory quantities are normalized using the same mean of demand employed for the normalization of demand quantities. Daily production capacities are set to the daily average of 110% of each week's demand and the target service level is 95% in all cases. Beginning on hand inventories are set to the initial safety stock plus the first ten days of demand during the training period. The model is run backwards through the first three weeks of training data to allow it to reach stable operation, although in general we find that the model settles down in 10 periods or fewer.

Figure 3-3. Inventory Comparison for CPU Line Item 1

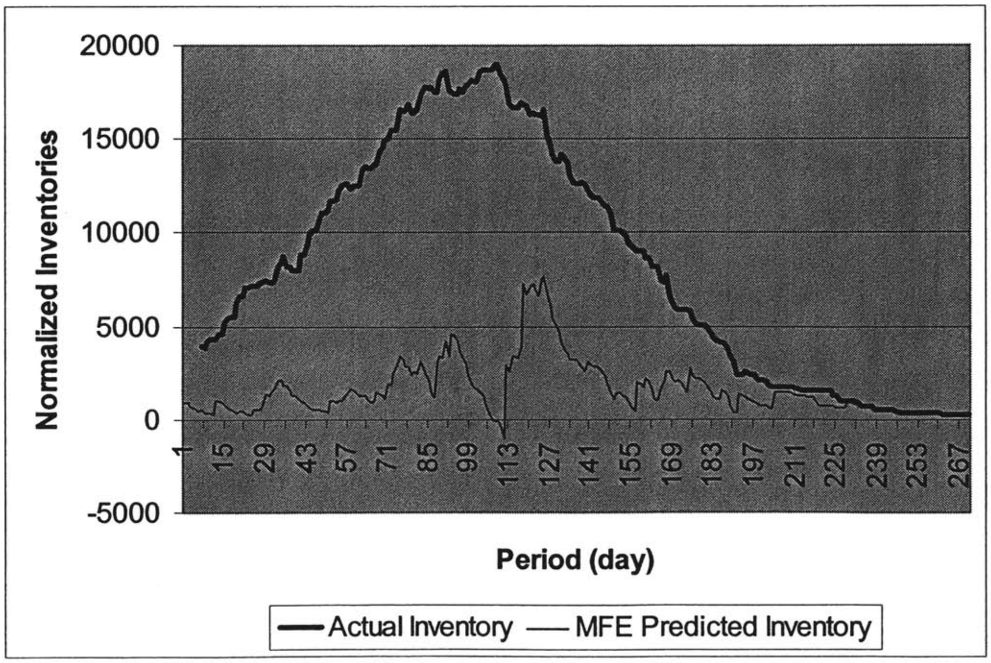


Figure 3-4. Inventory Comparisons for CPU Line Item 2

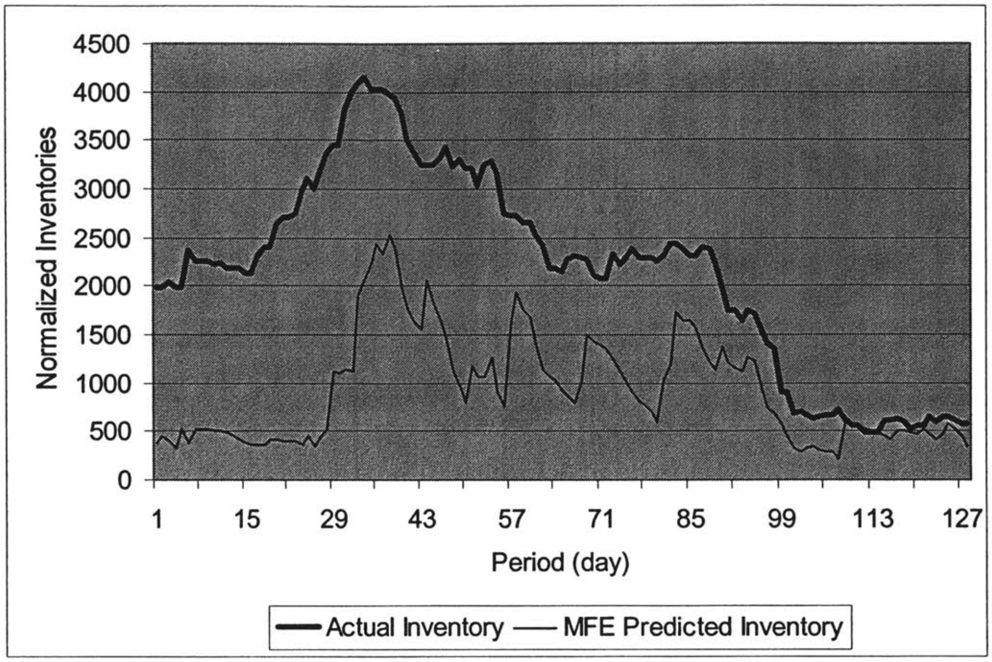


Figure 3-5. Inventory Comparison for CPU Line Item 3

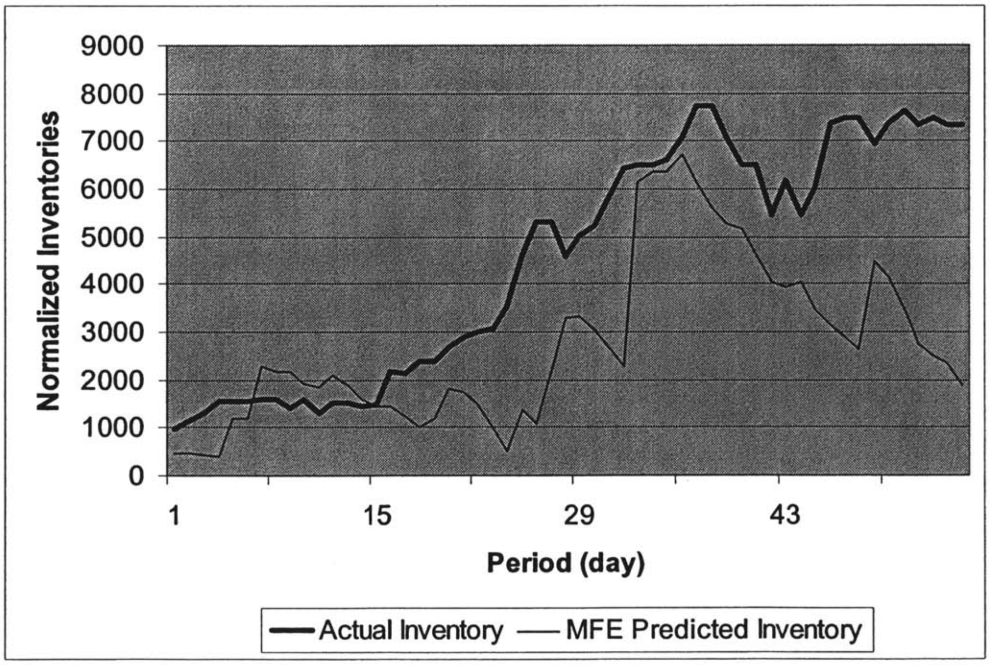


Figure 3-6. Inventory Comparison for CPU Line Item 4

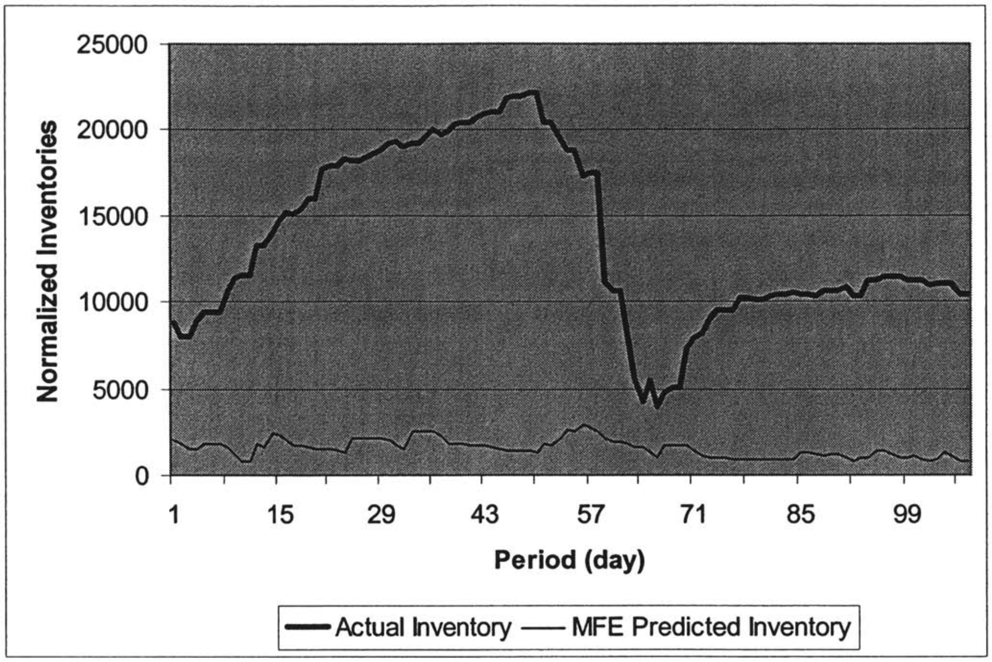


Table 3-1. Service Levels for MFE Inventory Simulations

CPU Line Item	Simulated Service Level
1	98.15 %
2	100.00 %
3	100.00 %
4	100.00 %

As shown in Table 3-1, service level goals are exceeded in all cases. The degree of excess service is larger than expected, perhaps due to our assumption of perfect correlation between seasonal forecast error and forecasted levels.

Table 3-2. Potential Inventory Reductions with Use of MFE[8]

CPU Line Item	Potential Inventory Reduction
1	80.2 %
2	57.1 %
3	39.2 %
4	88.5 %

The results of our historical simulations are dramatic. Table 3-2 illustrates the scale of reductions that could become possible should Intel fully implement this model. While several roadblocks currently exist, the quantity of these reductions provides a powerful source of motivation for implementation. We examine these roadblocks and other organizational factors in the next section.

[8] Note that these reductions rely on the assumed values for lead times, lead time variability, and forecast error. However, the values listed here are similar to those we have found when actual values are used for lead times. Forecast errors are still assumed in this case, although we believe them to be reasonable for the short forecast horizons we face in this application of the model.

4 ORGANIZATIONAL IMPACTS

In this section we examine the impact of implementing our proposed inventory model. An end-state vision of the new supply chain paradigm is presented, followed by a roadmap for its achievement. The section concludes with an examination of future improvements to our model.

4.1 IMPLEMENTATION STRATEGIES[9]

The conversion of Intel's supply chain from push to pull is far from trivial – it is a paradigm change in the process for managing Intel's entire manufacturing system. Changing the information process that generates production signals to Intel's factories also will require changes to performance measurements, organizational structures, and corporate culture.

Change on such a scale must be accomplished thoughtfully and incrementally – if the mechanics of the system were ready for implementation tomorrow, the effort would fail due to lack of buy-in from the organization. Even if buy-in existed, the effort would still fail due to the fact that it would be entirely new. In short, change of this magnitude cannot be expected to occur all at once – feedback, correction, and additional discovery are required.

4.1.1 Envisioning a Pull-Based Supply Chain

There are many possible paths to a pull-based supply chain paradigm. Of paramount importance, however, is that the entire organization be aware of the vision for the end-state and the roadmap to get there. In Intel's case, this end-state vision is a

[9] Many of the concepts in this section are drawn from Hammer [1995].

supply chain management process that generates its own production schedules based solely on demand forecasts, factory capacities, pricing, and historical data. Moreover, this process is a simple window into many of the metrics that drive the company's success: demand patterns, lead time patterns, forecast error, and production yields. Such visibility will allow Intel to reduce supply chain costs even further through the solution of as yet unknown problems previously covered up by inventory. Indeed, similar efforts at other major corporations have yielded dramatic results – Kodak has removed more than $500 million in inventory from its supply chain, and Proctor and Gamble has experienced similar results.[10]

4.1.2 Roadmaps for Changing Paradigms

Such a vision is indeed compelling, but what should be Intel's first step towards its realization? One possible roadmap could be to conduct a series of tests – historical simulations such as those presented in this thesis fall into this category. Offline simulations observed in real time, parallel to actual operations would come next, followed by an experimental pilot on a small set of actual products, then finally full rollout across all relevant products. While promising, this option is unfortunately currently impossible due to the problems with forecasts and lead times outlined in Chapter 3.

A roadmap that allows for these problems might entail the following:

1. Develop a process for the weekly assessment of lead times, lead time variability, and the impact they have on the company's bottom line. Re-

[10] Conversations with Earl Chapman, Eastman Kodak, and William Tarlton, Proctor and Gamble, September 2002.

align manufacturing performance measurements to include goals for lead time reduction and the mitigation of lead time variability. Shorter, more predictable lead times will result in the quick reduction of both finished goods and work in process, even without an automated replenishment system. Indeed, this is where Kodak reaped a large portion of its inventory reductions!

2. In parallel to the manufacturing effort, begin to generate shipment forecasts for a small set of test products. The products chosen should be the same as those actually used in live pilot programs. As such, these products should have stable demand and low production volumes. The forecasts need only cover shipments up to a horizon equal to the estimated lead time. Clear and visible support of the team performing this effort must be present from senior management for this and subsequent efforts to succeed. The team's boss and his or her boss must also be held accountable for their treatment of the project both during and after its completion.
3. With these forecasts, conduct an offline simulation of a replenishment system using weekly rather than daily quantities. Low production volumes allow capacity constraints to be neglected. Assume lead time characteristics equal to the goals set in step one. At this stage, use rules of thumb for setting inventory goals. A goal such as this might be two weeks of demand based on a four week moving average comprised of the previous two weeks' shipments and forecasts of the next two weeks'

shipments. These goals should be sufficiently high that stockouts are not a concern.

4. Correct any errors in the process, then conduct a pilot on the actual products when lead times are sufficiently under control.
5. Develop a system for monitoring forecast error over the lead time horizon.
6. In parallel to the pilot, conduct another offline simulation using a weekly model that estimates inventory targets mathematically, rather than experientially.
7. Develop a methodology and tool for setting capacity constraints across multiple products using unconstrained replenishment signals. This research could be conducted by a Class of 2004 MIT Leaders for Manufacturing intern.
8. In parallel to the development of this tool, transition the pilot to the weekly mathematical replenishment model.
9. Rollout the new system across other products, first using experiential inventory goals then transitioning to mathematical ones. Begin with the products that are easiest to manage and then move to the harder ones. Adjust performance measurements and the organizational structure as necessary to support the new supply chain management process.
10. Repeat 6 through 10 for roll out of the daily forecasting and inventory management system presented in this thesis.
11. Repeat 6 through 10 for roll out of additional features to the system such as other stages of the supply chain, financial optimizations, etc.

Throughout this entire roadmap, constant correction will be required and new challenges may be uncovered. Additionally, the transition of job descriptions should be gradual. Automation of manual processes and consolidation of effort should be attempted incrementally.

4.1.3 Summary of Recommendations for Implementation

In sum, a pull-based supply chain paradigm presents Intel with a vision that is compelling beyond the inventory reductions that would result from the use of our model. However, the achievement of that vision must be undertaken gradually with attention to factors such as performance drivers, organizational structure, culture, and politics. An incremental approach is vital to long-term success in order to allow for course corrections and to build momentum for change.

4.2 FUTURE MODELING POSSIBILITIES

Note that in the above roadmap, choosing how replenishment signals should be split amongst die inventory bins remains a manual process. A die inventory bin refers to a group of untested, unassembled chips grouped into similar speeds. This is the result of stochastic production processes – all chips produced on the same silicon wafer are not the same speed, but rather emerge along a distribution of different speeds. A replenishment signal for a 2.0 GHz product could be pulled into production from any die inventory bin containing chips of 2.0 GHz or greater. When combined with replenishment signals from other speeds, the optimal choice of die inventory bins becomes quite complex and requires the addition of financial considerations to the model. This process for choosing these bins could be automated as well if good methods can be created for doing so. However, an incremental roadmap for their introduction applies here as well.

An additional area for further development is the inclusion of production yields in the model. If these yields are predictable enough, a simple deterministic alteration to replenishments may be possible. If they are stochastic, more elaborate methods may be necessary.

A final suggestion for future study is the expansion of our single-stage multiplicative forecast error model to networks comprised of multiple stages. A detailed examination of bullwhip[11] effects resulting from the non-stationary nature of the model is warranted. Echelon stock policies could also be considered.

[11] The term bullwhip is used to characterize the effect of increasing demand volatility as one moves further away from the end customer.

5 CONCLUSIONS AND RECOMMENDATIONS

The scope of change required to fully implement this system is substantial and will require years to fully complete. The paradigm shift from push to pull must be accomplished incrementally and will necessitate corresponding changes in corporate culture, organizational structure, performance drivers, software, and infrastructure. However, the financial benefits are substantial: preliminary analysis of the net present value of inventory reductions exceeds $80 million, based on a three-year phased introduction across Intel's processor and flash memory product lines. In addition, when the system is implemented Intel will have the building blocks to an even leaner supply chain. This system gives the company a clear picture of the accuracy of their demand forecasts, the volatility of their demand, and the length and volatility of their lead times. With a more tightly controlled supply chain and the mathematics to describe it, financial optimization of service levels and product mix can be undertaken, as well as inventory reductions at other stages of the supply chain.

Several key insights can be drawn from this thesis:

- Periodic inventory replenishment models can be used for non-stationary processes with great success. While these models may not represent a lower bound on inventories, they provide a simple and powerful solution to an otherwise intractable problem.
- Multiplicative forecasting models offer a mathematically simple solution to non-stationary problems and provide substantial forecast error reduction.

- Models that can incorporate generational learning across products can make use of statistical models possible for short lifecycle products.
- Successful paradigm shifts in supply chain management techniques require open dissemination of the vision and roadmap for change. Many facets of an organization must be scrutinized to ensure that the plan will further align performance drivers, corporate strategy, and the structure of the supply chain. Practitioners should expect changes to be gradual rather than immediate.

6 REFERENCES

Axsäter, S., Rosling, K., "Notes: Installation vs. Echelon Stock Policies for Multilevel Inventory Control," Management Science 39-10, 1993.

Borchers, Michael, "Fitting an ARIMA Process to Data," Lecture Notes for Data Processing and Analysis, New Mexico School of Technology Geophysics Department, April 2001.

Brown, R.G., "Smoothing, Forecasting and Prediction of Discrete Time Series," Prentice-Hall: Englewood Cliffs, NJ, 1963.

Çakanyildirim, M., Roundy, R., "SeDFAM: Semiconductor Demand Forecast Accuracy Model," IEE Transactions 34, 2002.

Chambers, John C., Mullick, Satinder K., and Smith, Donald D., "How to Choose the Right Forecasting Technique," Harvard Business Review, July-August 1971.

Christensen, Clayton M., Raynor, Michael, and Verlinden, Matt, "Skate to Where the Money Will Be," Harvard Business Review, November 2001.

Fine, Charles H., "Clockspeed: Winning Industry Control in the Age of Temporary Advantage," Perseus Books: Reading, MA, 1998.

Gardner, E.S., "Exponential Smoothing: The State of the Art," Journal of Forecasting 4, 1985.

Godfrey, Gregory A., Powell, Warren B., "Adaptive Estimation of Daily Demands with Complex Calendar Effects for Freight Transportation," Transportation Research Part B 34, 2000.

Hammer, Michael, "The Reengineering Revolution: A Handbook," HarperBusiness, 1995.

Holt, C.C., Modigliana, F., Muth, J.F., Simon, H.A., "Planning Production, Inventories, and Work Force," Prentice-Hall: Englewood Cliffs, NJ, 1960.

Murty, Katta G., "Supply Chain Management in the Computer Industry," Working Paper, University of Michigan Department of Industrial and Operations Engineering, 2000.

Porter, Michael E., "How Competitive Forces Shape Strategy," Harvard Business Review, March-April 1979.

Silver, E.A., Peterson, R., "Decision Systems for Inventory Management and Production Planning," 2nd Edition, John Wiley & Sons, New York, 1985.

Synder, Ralph, "Forecasting Sales of Slow and Fast Moving Inventories," European Journal of Operational Research 140, 2002.

Winters, P.R., “Forecasting Sales by Exponentially Weighted Moving Averages,” Management Science 6, 1960.

APPENDIX A – NORMALIZED DEMAND DATA

The data in the following table have been normalized such that the mean of the training period data (see Table 2-1 for a list of training periods) is 100. Period 1 corresponds to a different starting date for each CPU Line Item.

Table A-1. Normalized Demand Data for CPU Line Items 1 through 4

	Normalized Demand for CPU Line Item			
Period (day)	1	2	3	4
1	134	0	7	0
2	113	0	23	502
3	133	112	56	56
4	65	40	0	1
5	57	0	0	0
6	199	279	5	2
7	280	0	8	8
8	134	0	211	334
9	113	0	256	650
10	133	28	283	14
11	65	0	138	0
12	57	64	43	196
13	199	13	390	180
14	280	80	266	7
15	40	0	17	195
16	5	8	13	458
17	40	12	395	4
18	30	0	25	0
19	33	4	0	244
20	2	7	72	0
21	232	20	45	0
22	79	0	435	0
23	216	1	635	120
24	0	41	258	143
25	153	54	52	0
26	3	158	560	0
27	10	59	438	0
28	60	28	163	10
29	86	17	330	5
30	3	11	265	233
31	0	19	483	383
32	36	16	276	1
33	14	186	108	1
34	60	410	320	0
35	108	0	182	9
36	17	16	327	1

	Normalized Demand for CPU Line Item			
Period (day)	1	2	3	4
37	47	228	824	691
38	0	36	318	29
39	8	269	337	39
40	76	565	54	14
41	381	165	1036	94
42	252	133	168	0
43	188	8	596	0
44	376	228	844	209
45	56	235	267	0
46	63	126	438	71
47	212	461	13	0
48	25	250	502	0
49	104	223	186	150
50	240	100	454	2
51	75	146	1117	100
52	7	79	323	9
53	9	110	200	0
54	37	543	94	28
55	66	183	809	114
56	95	127		1
57	62	31		459
58	9	343		261
59	27	26		446
60	222	114		0
61	36	441		0
62	5	107		288
63	238	26		1
64	190	37		55
65	169	198		589
66	14	73		74
67	207	86		0
68	39	273		1
69	62	111		3
70	258	52		170
71	242	19		371
72	74	49		196
73	117	164		35
74	43	88		0
75	195	96		86
76	45	141		3
77	192	36		10
78	148	63		64
79	201	89		0
80	25	111		0
81	127	83		0
82	230	72		0
83	134	49		5
84	574	135		0

	Normalized Demand for CPU Line Item			
Period (day)	1	2	3	4
85	985	31		15
86	100	116		119
87	91	295		62
88	194	174		23
89	104	134		7
90	167	195		11
91	865	122		425
92	570	13		14
93	515	36		6
94	132	89		10
95	0	215		13
96	274	369		17
97	370	34		321
98	1259	161		28
99	485	64		13
100	269	199		11
101	87	40		349
102	539	30		51
103	517	49		0
104	207	27		12
105	518	40		504
106	598	21		24
107	420	100		49
108	67	64		
109	0	36		
110	217	30		
111	318	31		
112	499	35		
113	491	0		
114	190	49		
115	10	18		
116	844	45		
117	296	60		
118	976	44		
119	545	6		
120	257	1		
121	41	53		
122	0	60		
123	177	49		
124	197	84		
125	232	62		
126	751	49		
127	291	79		
128	197	139		
129	338			
130	416			
131	983			
132	770			

	Normalized Demand for CPU Line Item			
Period (day)	1	2	3	4
133	852			
134	734			
135	558			
136	120			
137	560			
138	585			
139	184			
140	427			
141	0			
142	72			
143	31			
144	495			
145	227			
146	534			
147	71			
148	283			
149	47			
150	4			
151	205			
152	275			
153	211			
154	428			
155	654			
156	27			
157	0			
158	167			
159	192			
160	59			
161	420			
162	217			
163	106			
164	3			
165	276			
166	97			
167	261			
168	116			
169	878			
170	163			
171	0			
172	202			
173	165			
174	275			
175	645			
176	263			
177	4			
178	61			
179	257			
180	68			

	Normalized Demand for CPU Line Item			
Period (day)	1	2	3	4
181	454			
182	401			
183	338			
184	42			
185	0			
186	134			
187	69			
188	364			
189	266			
190	200			
191	36			
192	0			
193	123			
194	152			
195	433			
196	511			
197	126			
198	9			
199	5			
200	155			
201	126			
202	11			
203	92			
204	154			
205	97			
206	0			
207	51			
208	148			
209	90			
210	19			
211	30			
212	1			
213	0			
214	14			
215	1			
216	6			
217	103			
218	34			
219	19			
220	0			
221	65			
222	10			
223	1			
224	1			
225	486			
226	0			
227	0			
228	0			

	Normalized Demand for CPU Line Item			
Period (day)	1	2	3	4
229	1			
230	5			
231	51			
232	3			
233	0			
234	0			
235	1			
236	1			
237	235			
238	7			
239	24			
240	45			
241	0			
242	9			
243	5			
244	115			
245	50			
246	19			
247	0			
248	0			
249	0			
250	95			
251	20			
252	43			
253	15			
254	0			
255	0			
256	6			
257	0			
258	0			
259	1			
260	28			
261	0			
262	0			
263	4			
264	0			
265	11			
266	16			
267	4			
268	0			
269	3			
270	4			
271	10			
272	51			
273	0			
274	0			
275	0			
276	0			

	Normalized Demand for CPU Line Item			
Period (day)	1	2	3	4
277	22			